
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Informační technologie

Řadič paměťové karty

Memory Card Controller

Diplomová práce

Autor: **Bc. Tomáš Kalina**

Vedoucí práce: doc. Ing. Milan Kolář, CSc.

Konzultant: Ing. Ondřej Zelinka, Ph.D.

V Liberci 28. 8. 2011

Originální zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, která vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 28. 8. 2011

Podpis:

Abstrakt

Tato diplomová práce se zabývá využitím programovatelného hradlového pole jako řadiče paměťové karty SD.

Řadič byl navržen tak, aby umožňoval automatické zapisování hodnot. Ty mu může dodávat například měřicí zařízení, nahrané ve stejném hradlovém poli. Je navržen jednoduchý souborový systém, kdy je na kartě uchovávána hodnota další volné adresy.

Pro praktickou realizaci byla použita deska FUB, obsahující čip Altera Cyclone III a dostatečný počet vstupních a výstupních prvků. Z různých paměťových karet byl vybrán typ Secure Digital, konkrétně Emgeton Flexaret o kapacitě 2 GB.

Pro odzkoušení řadiče byl dále použit vstupní obvod, generující potřebné množství dat. Zobrazování vyčtených údajů řeší výstupní obvod pomocí dvou sedmissegmentových displejů.

Klíčová slova: SD paměťová karta, řadič, FPGA, Altera Cyclone III, měření, ukládání dat

Abstract

This diploma thesis discusses the use programmable gate array as a SD card controller.

The controller was designed to allow automatic recording of values. These data can supply for example measuring device recorded in the same FPGA. It is designed simple file system, the value of the next free addresses is stored on the card.

For practical implementation was used FUB board, including Altera Cyclone III chip and adequate number of input and output elements. From various memory card types was selected Secure Digital, namely Emgeton Flexaret with capacity 2 GB.

The input circuit, generating the necessary amount of data was used to test the controller. Viewing received data solves output circuit using two seven-segment displays.

Keywords: SD Memory Card, controller, FPGA, Altera Cyclone III, measurement, data storage

Obsah

Prohlášení	3
Abstrakt	4
Obsah	5
Seznam obrázků	8
Seznam tabulek	9
Seznam použitých symbolů a zkratek	10
1 Úvod	11
2 Paměťové karty	12
2.1 Historie	12
2.2 Dostupné karty	13
2.2.1 Compact Flash (CF)	13
2.2.2 Memory Stick (MS)	13
2.2.3 Extreme Digital (xD)	14
2.2.4 MultiMedia Card (MMC)	15
2.2.5 Secure Digital (SD)	15
2.3 Porovnání	16
2.4 Nejvhodnější karta	17
3 Jazyk VHDL	18
3.1 Historie	18
3.2 Syntaxe	18
3.3 Některé zásady popisu ve VHDL	19
3.4 Altera Quartus II 9.1 Web Edition	19
4 Obvody FPGA	20
4.1 Vývojová deska FUB	20

5	SPI sběrnice	21
5.1	Polarita a fáze hodin	21
6	Komunikace s kartou.....	22
6.1	Formát příkazů pro SD kartu	22
6.2	Formáty odpovědí od SD karty.....	22
6.2.1	Odpověď R1	22
6.2.2	Odpověď R1b	23
6.2.3	Odpověď R2	23
6.2.4	Odpověď R3	24
6.2.5	Odpovědi R4 a R5	24
6.2.6	Odpověď R7	24
6.3	Inicializace SD karty do SPI režimu.....	25
6.4	Režim přenosu	25
6.5	Nastavení velikosti bloku dat.....	25
6.6	Zápis na SD kartu	26
6.6.1	Data response token	26
6.7	Čtení z SD karty.....	27
6.7.1	Chybový token při chybě čtení	27
6.8	Startovací token při přenosu dat	28
7	Řadič paměťové karty	29
7.1	Stavový automat inicializace SD karty	30
7.2	Stavový automat pro zjištění nejbližší volné adresy.....	31
7.3	Stavový automat zápisu na SD kartu	32
7.4	Stavový automat kontroly zapsání dat	34
7.5	Stavový automat aktualizace volné adresy	35
7.6	Stavový automat čtení z SD karty.....	35
7.7	Dělička frekvence	36

7.8	Zpoždění začátku inicializace	36
7.9	Obvody vstupu a výstupu	36
7.9.1	Obvod vstupních dat	37
7.9.2	Obvod výstupních dat	38
8	Simulace a odzkoušení.....	39
8.1	Simulace.....	39
8.2	Odzkoušení	40
	Závěr	41
	Přílohy.....	42
	Příloha A	42
	Literatura	43

Seznam obrázků

Obr. 2.1 - Paměťová karta Compact Flash	13
Obr. 2.2 - Paměťová karta Sony MemoryStick	14
Obr. 2.3 - Paměťová karta Extreme Digital	14
Obr. 2.4 - Paměťová karta MultiMedia Card.....	15
Obr. 2.5 - Paměťová karta Secure Digital.....	16
Obr. 4.1 - FPGA University Board	20
Obr. 5.1 - Časový digram ukazující polaritu a fázi hodin.....	21
Obr. 6.1 - Posloupnost zápisu na SD kartu, převzato z [1].....	26
Obr. 6.2 - Posloupnost čtení z SD karty, převzato z [1]	27
Obr. 6.3 - Chyba při čtení z SD karty, převzato z [1]	27
Obr. 7.1 - Schéma zapojení řadiče	29
Obr. 7.2 - Obecný stavový automat komunikace s SD kartou.....	29
Obr. 7.3 - Stavový automat pro inicializaci SD karty.....	31
Obr. 7.4 - Stavový automat pro zjištění nejbližší volné adresy	32
Obr. 7.5 - Stavový automat zápisu na SD kartu.....	33
Obr. 7.6 - Stavový automat kontroly zapsání dat	34
Obr. 7.7 - Stavový automat čtení z SD karty	35
Obr. 7.8 - Stavový automat obvodu vstupních dat	37
Obr. 7.9 - Stavový automat obvodu výstupních dat	38

Seznam tabulek

Tab. 2.1 - Porovnání paměťových karet	16
Tab. 6.1 - Formát příkazu	22
Tab. 6.2 - Formát odpovědi R1	22
Tab. 6.3 - Formát odpovědi R2	23
Tab. 6.4 - Formát odpovědi R3	24
Tab. 6.5 - Formát odpovědi R6	24
Tab. 6.6 - Formát Data response tokenu	26
Tab. 6.7 - Formát chybového tokenu při chybě čtení	27
Tab. 6.8 - Formát start bytu ve startovacím tokenu	28

Seznam použitých symbolů a zkratk

CC	Card Controller - vnitřní řadič karty
CPHA	Clock Phase - fáze hodinového signálu
CPOL	Clock Polarity - polarita hodinového signálu
CRC	Cyclic Redundancy Code - cyklický redundantní kód
ECC	Error Correction Code - kontrola a oprava chyb
EEPROM	Electrically Erasable Programmable Read-Only Memory - elektricky mazatelná, programovatelná paměť
FPGA	Field Programmable Gate Array - programovatelné hradlové pole
FUB	FPGA University Board - přípravek s FPGA
IEEE	Institute of Electrical and Electronics Engineers
JTAG	Joint Test Action Group - komunikační a testovací rozhraní
LED	Light-Emitting Diode - svítivá dioda
LSB	Least Significant Bit - nejméně významný bit
MISO	Master In Slave Out - datový vodič sběrnice SPI
MOSI	Master Out Slave In - datový vodič sběrnice SPI
MSB	Most Significant Bit - nejvíce významný bit
OCR	Operation Conditions Register - registr provozních podmínek
SCLK	Serial Clock - hodinový vodič sběrnice SPI
SD	Secure Digital - typ paměťové karty
SDHC	SD High Capacity - SD s kapacitou až 32 GB
SDIO	SD Input Output - SD se vstupně výstupními obvody
SDXC	SD Extreme Capacity - SD s kapacitou až 2 TB
SPI	Sertial Peripheral Interface - sériové komunikační rozhraní pro periferní zařízení
SS	Slave Select - výběrový vodič sběrnice SPI
VHDL	Very High-Speed Integrated Circuit Description Language - jazyk pro popis velmi rychlých integrovaných obvodů

1 Úvod

Různá měření sebou přináší potřebu naměřená data ukládat. Někdy je ale nepraktické a občas i nemožné mít připojené PC. Samozřejmě je možné měřicí zařízení připojit do počítačové sítě, ale ani to nemusí být ekonomicky únosné. Nevyplatí se to ani v případě, že data potřebujeme znát pouze v dlouhých časových intervalech. V případě, že se měřicí obvod nachází v obvodu FPGA, je možnost vytvořit paměťové místo přímo v něm. Ale i v tomto případě je nutné pro vyčtení dat odnést obvod s sebou nebo připojit počítač. V případě využití řadiče a paměťové karty lze data pohodlně odnést a případně vyměnit plnou kartu za prázdnou.

Ukládání dat na paměťové karty je dnes velmi oblíbené. Hlavními důvody jsou nízká cena, rostoucí dostupná kapacita a kompaktnost. Proto se přímo nabízí využití paměťových karet u měřicích přístrojů.

Pokud stačí mít data jednoduše uložena na kartě, není potřeba žádné složité zařízení. Postačí jednoduchý řadič, který umí automaticky zapsat na další volnou adresu, přečíst poslední zapsaná data a zobrazit je na displeji, či vymazat data na kartě.

Tato diplomová práce se zabývá právě takovým řadičem, vytvořeným programovatelným hradlovým polem a popsáním v jazyce VHDL. Místo měřicího zařízení je k němu připojen generátor dat, na výstupu je obvod, který poslední zapsaná data zobrazí na displeji. Není problém místo těchto obvodů připojit jiná zařízení, pokud splní podmínky, které řadič má, když čte nebo posílá data.

2 Paměťové karty

V dnešní době jsou mezi lidmi velmi rozšířena zařízení, která vyžadují ukládání vcelku velkých objemů dat. Ať jde o digitální fotoaparáty, videokamery, hudební přehrávače či mobilní telefony, většinou již nedostačují integrované paměťové prostory. Proto dnes většina těchto přístrojů umožňuje použití paměťových karet. Jejich základem je paměť EEPROM nebo flash paměť. Ta nevyžaduje pro zachování informace stálé napájení, data v ní zůstanou až několik desítek let. Nepopíratelnou výhodou je také absence mechanických částí náchylných k mechanickému poškození.

2.1 Historie

První paměťovou kartou byl v roce 1992 Flash Disk firmy SanDisk založený na rozhraní PCMCIA, ale pro svoje velké rozměry byl použitelný maximálně v noteboocích. V roce 1994 se na trhu objevil nový formát CompactFlash, který již splnil požadavky nastupující vlny digitálních fotoaparátů. První kartou, která v sobě neobsahovala řadič, byla o rok později Smart Media.

S nástupem funkcí v mobilních telefonech a jejich požadavků na paměťový prostor, byl vyvinut nový typ karty, MultiMedia Card (MMC). Nástupcem pak byla karta Secure Digital (SD), která přinesla při takřka stejných rozměrech větší kapacity a hlavně digitální ochranu obsahu. SD karta se dočkala několika nových verzí, zvyšujících kapacitu a rychlosti i snižujících fyzickou velikost.

Velmi rozšířenou kartou byl také Memory Stick od firmy Sony, která do svých zařízení zpočátku ani jiné typy nepoužívala.

Jedním z posledních typů karet je xD (Extreme Digital). Rozměrově jde o jednu z nejmenších karet.

V posledních letech se nové typy na trhu příliš neobjevují, většinou jde o vylepšení starších, již rozšířených formátů. Dá se říci, že dnešním standardem pro použití v mobilních telefonech, digitálních fotoaparátech nebo hudebních přehrávačích je karta SD. Dokonce i firma Sony již začala u některých zařízení přecházet na tento typ. V nejlepších digitálních zrcadlovkách převažuje stále karta CompactFlash.

2.2 Dostupné paměťové karty

2.2.1 Compact Flash (CF)

Karty Compact Flash (obr. 2.1) jsou založeny na technologii PATA (Parallel ATA). Jsou používány především profesionály, ať již k záznamu fotek, tak i videí. Jsou také využívány v průmyslu, kde, díky absenci mechanických částí a nízké spotřebě, nahrazují pevné disky.

Existují dva typy CF karet, fyzicky se liší pouze šířkou. Type I : 43×36×3,3 mm, Type II : 43×36×5 mm. Type II již ale pro uchování dat nepoužívá pouze technologii Flash, ale také microdrive. V tom případě jde v podstatě o jednopalcový pevný disk. Karta má 50 pinů. Na trhu je dnes k dostání s kapacitou 128 GB.

Od roku 1994 bylo vydáno 6 revizí, které postupně zvýšili rychlost i kapacitu:

- Revize 1.0 (1995), rychlost 8.3 MB/s, podporovaná kapacita až 128 GiB (137 GB).
- Revize 2.0 (2003), rychlost 16.6 MB/s.
- Revize 3.0 (2004), podpora přenosové rychlosti až 66 MB/s, zabezpečení heslem.
- Revize 4.0 (2006), maximální rychlost přenosu 133 MB/s.
- Revize 5.0 (2010), přidáno 48-bitové adresování (podpora kapacity až 128 Petabytů).
- Revize 6.0 (listopad 2010), přenosová rychlost až 167 MB/s.



Obr. 2.1 - Paměťová karta Compact Flash

2.2.2 Memory Stick (MS)

Karty Memory Stick (obr. 2.2) vyrábí od roku 1998 firma Sony. Vyráběly se ve třech variantách rozměrů. Největší je původní MS (50×21,5×2,8 mm), ten se již dnes nevyrábí. Menší varianta se jmenuje Memory Stick Duo (31×20×1,6 mm) a nejmenší

Memory Stick Micro (15×12,5×1,2 mm). Každá z těchto rozměrových variant existuje ještě jako rychlejší verze s označením PRO. Komunikace probíhá přes 10 pinů.

Původní MS měl maximální kapacitu 128 MB, MS Duo je dnes na trhu s kapacitou až 32 GB, MS Micro 16 GB. Memory Stick PRO má rychlost 20 MB/s oproti 2,5 MB/s u standardního Memory Sticku.



Obr. 2.2 - Paměťová karta Sony MemoryStick

2.2.3 Extreme Digital (xD)

Karta xD (obr. 2.3) vyvinutá firmami Olympus a Fujifilm v roce 2002 jako náhrada za Smart Media. Podobně jako MS od Sony byla využívána v podstatě pouze ve výrobcích těchto firem. První generace byla omezena kapacitou 512 MB, to se změnilo po vydání různých variant. Jsou tři: M, H a M+. Změnila se struktura, tudíž již nejsou plně kompatibilní se staršími přístroji. Maximální teoretická kapacita je u těchto nových typů až 8 GB, ale k dostání jsou maximálně s kapacitou 2 GB. Přenosová rychlost (max. 6 MB/s) i kapacita je v porovnání s ostatními formáty karet nízká a cena vysoká, proto ani jejich obliba není nijak velká.

Oba výrobci digitálních zařízení již upouští od používání pouze jednoho formátu karet (xD) a jejich výrobky tak lze používat i s jiným typem karet.



Obr. 2.3 - Paměťová karta Extreme Digital

2.2.4 MultiMedia Card (MMC)

MMC (obr. 2.4) byla představena v roce 1997 společnostmi Siemens AG a SanDisk. Ve své době šlo o nejmenší paměťovou kartu na trhu ($32 \times 24 \times 1,4$ mm). Maximální teoretická kapacita karet MMC je 128 GB, ale nejvyšší prodávaná je 8 GB. Protože byla přenosová rychlost v porovnání s ostatními formáty nízká (2 MB/s), byla v roce 2005 představena MMCplus. Její teoretická rychlost je 52 MB/s. Kvůli možnosti použití MMC karet i ve stále se zmenšujících mobilních telefonech a podobných zařízeních, byly vyvinuty MMCmobile (2003) a MMCmicro (2005).



Obr. 2.4 - Paměťová karta MultiMedia Card

2.2.5 Secure Digital (SD)

Od roku 1999 byla SD karta (obr. 2.5) vyvíjena jako náhrada za MultiMedia Card. Secure Digital se poté díky svému rozšíření a cenám stala v podstatě průmyslovým standardem.

SD karty mají stejné rozměry jako MMC. Čtečky SD tak komunikují s kartami MMC, ale opačně to neplatí. V roce 2003 byla vyvinuta menší verze, miniSD ($21,5 \times 20 \times 1,4$ mm), o dva roky později ještě menší microSD ($11 \times 15 \times 1$ mm).

Původní SD karta měla maximální kapacitu 4 GB, rychlost 25 MB/s. V roce 2006 přišla asociace SD s kartou SDHC (High-Capacity). Ta zvýšila kapacitu až na 32 GB (omezeno specifikací SD 2.0). I ta má tři typy velikostí.

Ještě větší kapacitu (teoreticky až 2 TB) má SDXC (Extended Capacity) z roku 2009. Zde byl použit souborový systém exFAT oproti FAT32 u předchozích verzí.

Zajímavým řešením jsou i zařízení SDIO (Secure Digital Input Output). Z jedné strany je standardní rozhraní SD karty, z druhé vstupně výstupní zařízení. Jedná se o GPS, Wi-Fi, digitální kameru, Bluetooth nebo například čtečku otisků prstů.



Obr. 2.5 - Paměťová karta Secure Digital

2.3 Porovnání

V tabulce 2.1 jsou porovnány karty dostupné v českých internetových obchodech. Cena byla zjištěna přes porovnávač cen na adrese heureka.cz dne 29.8.2011. Vždy byla vybrána karta s nejvyšší dostupnou kapacitou, v případě více takových karet, ta s nejvyšší rychlostí.

Tab. 2.1 - Porovnání paměťových karet

Typ karty	Dostupná kapacita v GB	Rychlost zápisu v MB/s	Přibližná cena v Kč	Přibližná cena za 1 GB
CF	128	45	10 000,-	78,-
MS	32	30	2 600,-	81,-
xD	2	3,75	500,-	250,-
MMC	4	22	450,-	113,-
SD	4	25	210,-	53,-
SDHC	32	min. 10	900,-	28,-
SDXC	64	min. 16	2 000,-	31,-

2.4 Nejvhodnější karta

Jak lze vyčíst z tabulky 2.1, vybrání nejvhodnějšího typu paměťové karty závisí na jejím budoucím použití. Při velkém objemu dat a vysoké rychlosti jejich zaznamenávání (např. sekvenční snímání fotografií s velkým rozlišením) je rozhodně nejvhodnější karta typu Compact Flash. Takovým požadavkům ovšem také odpovídá vysoká cena.

Pro použití paměťové karty jako záznamového média při měření lze očekávat vyšší rychlost zápisu, ale ne tak objemná data. V tom případě se tedy jeví jako nejlepší volba karta SDHC, případně SD, pokud by stačila kapacita do 4 GB.

Další důležitá informace je, zda jsou volně dostupné informace o řadiči karty. Compact Flash, Memory Stick, xD ani MultiMedia Card otevřený standard nemají. Secure Digital ano, pouze s výjimkou na zabezpečené části karty.

Po zvážení všech těchto kritérií byla vybrána jako nejvhodnější právě karta Secure Digital.

3 Jazyk VHDL

3.1 Historie

Návrh řadiče je proveden pomocí popisného jazyka VHDL (**V**ery **H**igh **S**peed **I**ntegrated **C**ircuits **H**ardware **D**escription **L**anguage - popisný jazyk hardwaru velmi rychlých integrovaných obvodů).

V roce 1986 byl jazyk uvolněn organizaci IEEE (**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers). V roce 1987 byl poprvé publikován standard jazyka VHDL. Největší změnou bylo v roce 1999 přidání podpory VHDL-AMS (analogové a smíšené signály).

Obvody popsané jazykem VHDL jsou většinou plně převoditelné do hardwaru (tzv. syntetizovatelné), ale stále existují příkazy, u kterých to nejde. Syntetizovatelné nejsou také testovací programy (testbench), ty slouží pouze k odzkoušení obvodu v simulaci.

Velkou výhodou je nezávislost tohoto jazyka na obvodu, do kterého bude výsledný návrh implementován. Při návrhu v podstatě není nutné zabývat se typem cílového obvodu či jeho výrobcem. Výhodná je také forma otevřeného standardu, která umožňuje použití pro nás uživatelsky přívětivějšího programovacího prostředí od různých firem.

3.2 Syntaxe

Každý obvod musí být popsán pomocí entity a alespoň jedné architektury. Entita je v podstatě vnější popis obvodu. Jsou v ní definovány vstupní i výstupní porty i konstanty používané v rámci entity. U portů je definován režim přenosu dat (vstupní, výstupní, vstupně-výstupní, buffer). Také se u portu určuje jeho datový typ (výčtový, celočíselný, reálný, atd.).

V architektuře je popsáno chování entity. Jde například o vztahy mezi porty, funkce či procedury. Toto chování může být pro více architektur jedné entity různé (jednotné pouzdro, ale různé obvody uvnitř).

3.3 Některé zásady popisu ve VHDL

Je dobré si návrh předem rozdělit na více jednodušších bloků, které následně pospojujeme dohromady. Zjednoduší to práci a hlavně následné opravování případných chyb. Také je vhodné oddělit od sebe bloky s řízením toku dat a samotná data.

Je také vhodné použít globální reset, pomocí kterého jsme schopni dostat obvod do počátečního stavu.

Samozřejmostí by mělo být používání komentářů, které umožňují lepší čitelnost kódu nejen pro autora, ale také pro ostatní. Pro lepší čitelnost je dobré klíčová slova psát velkými písmeny. Větší přehlednost přináší i úplné zapisování tam, kde lze vybrat i zkratkovitě (např. u Port Map preferovat jmenné přiřazování před polohovým).

3.4 Altera Quartus II 9.1 Web Edition

Pro tvorbu kódu, překlad a syntézu i pro nahrání do FPGA byl použit program Quartus od firmy Altera. Jde o verzi 9.1, pro studenty volně dostupnou na internetu.

Quartus splňuje hlavní požadavky pro práci s FPGA. V jednom programu tak lze pohodlně vytvářet kód, provádět analýzu a syntézu i simulace.

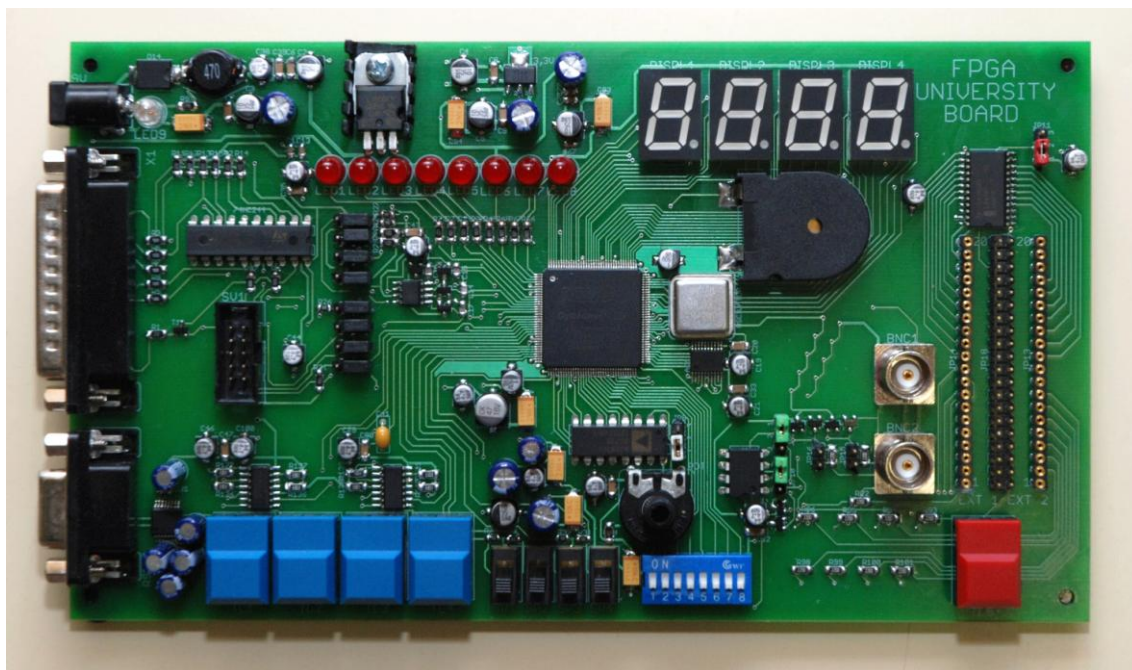
4 Obvody FPGA

Programovatelná hradlová pole (Field Programmable Gate Array) jsou speciální integrované obvody, které obsahují velké množství programovatelných bloků, které lze propojit pomocí spojů podle požadavků. Největší výhodou je jejich univerzálnost. Výsledná funkce obvodu je dána až zákazníkem. Dají se proto použít například u malých sérií obvodů. Plusem je samozřejmě i přeprogramovatelnost. Dnes již cena těchto obvodů klesla na přijatelnou mez. Největšími výrobci jsou Altera a Xilinx.

4.1 Vývojová deska FUB

Obvod FPGA použitý pro ověření funkčnosti je umístěn na desce FUB (FPGA University Board), vyrobené na fakultě mechatroniky. Deska obsahuje hradlové pole Altera Cyclone III, EP3C5E144C8. Dále jsou zde A/D a D/A převodník, digitální potenciometr, krystalový oscilátor, piezoměnič, 8 LED diod, 4 sedmissegmentové displeje, 1 neošetřené tlačítko, 4 ošetřená, posuvné přepínače a kódový spínač DIP08. Rozmístění těchto součástek je vidět na obr. 4.1.

Jako programovací rozhraní slouží paralelní port nebo JTAG; dále je zde sériový port RS232 (pro komunikaci FPGA po sériové lince), dutinkové lišty (piny FPGA, D/A, A/D převodník), 2 BNC konektory (lze připojit generátor signálu či osciloskop).



Obr. 4.1 - FPGA University Board

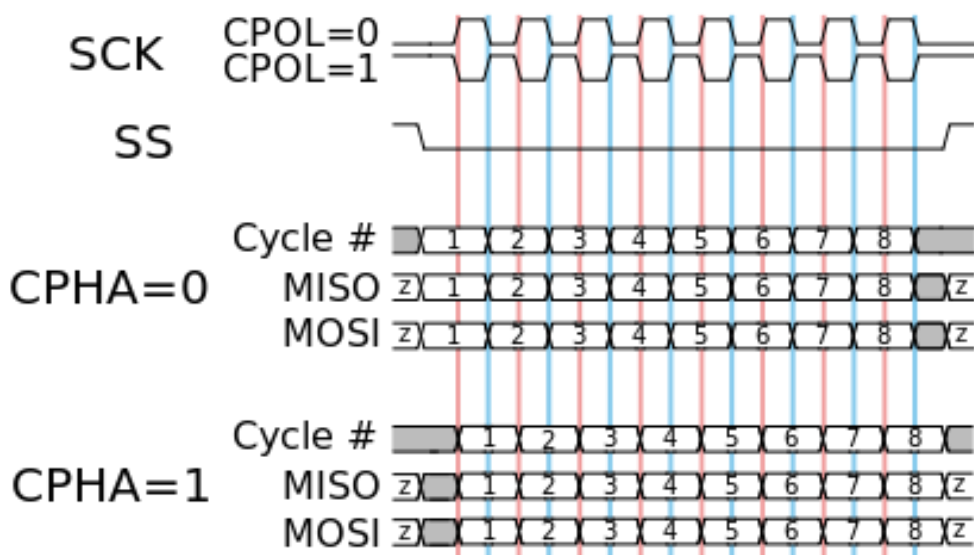
5 SPI sběrnice

SPI (Serial Peripheral Interface) sběrnice slouží k synchronní komunikaci po sériové lince. Funguje v plně duplexním režimu, data jsou tedy zároveň odesílána i přijímána. Jedno zařízení je označeno jako Master, ostatní jako Slave. Master řídí celou komunikaci, generuje hodinový signál a vybírá, kterému zařízení typu Slave patří data.

Komunikace probíhá po 4 vodičích. Po *MOSI* (Master Output, Slave Input) vysílá Master postupně data po 1 bitu. Přes *MISO* (Master Input, Slave Output) stejným způsobem přijímá data ze Slave zařízení. Na vodič *SCLK* (Serial Clock) vysílá Master hodinový signál. Přivedením vodiče *SS* (Slave Select) do logické nuly je vybráno příslušné zařízení. V případě, že komunikujeme pouze s jedním přijímačem, může být tento signál stále nastaven do logické nuly.

5.1 Polarita a fáze hodin

Protože jsou čtyři kombinace polarity a fáze hodin, je potřeba určit, kdy budou platná data přítomna na sběrnici. Polarita hodin značí, zda bude první platná hrana hodin náběžná či sestupná. Fáze říká, zda jsou data platná na první nebo druhou hranu hodin. Na obrázku 5.1 je fáze označena jako *CPHA* a polarita jako *CPOL*.



Obr. 5.1 - Časový diagram ukazující polaritu a fázi hodin

U paměťové karty SD je $CPOL = 0$ a $CPHA = 0$. To znamená, že na náběžnou hranu hodin *SCLK* musí být na datových vodičích platná data a na sestupnou hranu se na datové vodiče přesunou následující data.

6 Komunikace s paměťovou kartou

6.1 Formát příkazů pro SD kartu

Každý příkaz má jednotnou délku 6 bytů, tj. 48 bitů. Nejdříve se přenáší nejvýznamnější bit (MSB) a naposledy nejméně významný (LSB). Významy jednotlivých bitů lze vyčíst z tabulky 6.1. Šest bitů indexu příkazu udává, který příkaz z tabulky je odesílán, k dispozici máme 64 možných variant.

Tab. 6.1 - Formát příkazu

Popis	Start bit	Přenosový bit	Index příkazu	Argument	CRC7	Stop bit
Číslo bitu	47	46	45 až 40	39 až 8	7 až 1	0
Hodnota	'0'	'1'	x	x	x	'1'

6.2 Formáty odpovědí od SD karty

Na příkazy SD karta odpovídá v SPI režimu jedním z pěti možných druhů odpovědí.

6.2.1 Odpověď R1

Odpověď R1 má délku jednoho bytu. Je odeslán po každém příkazu, kromě těch, které požadují status karty. První bit je vždy nastaven do logické nuly.

Tab. 6.2 - Formát odpovědi R1

7	6	5	4	3	2	1	0
'0'	Parameter error	Address error	Erase sequence error	Com CRC error	Illegal command	Erase reset	In idle state

Pokud je některý z následujících bitů (tab. 6.2) v logické jedničce, značí to:

- Parameter error: argument příkazu mimo rozsah,
- Address error: neodpovídající adresa,
- Erase sequence error: chyba v sekvenci příkazů mazání,
- Communication CRC error: chyba CRC posledního příkazu,
- Illegal command: neplatný příkaz,
- Erase reset: mazací sekvence přerušena,
- In idle state: karta provádí inicializační sekvenci.

6.2.2 Odpověď R1b

Formát odpovědi R1b je stejný jako R1, navíc ještě je generován signál busy na datové lince. Jde o byty o hodnotě 0x00h, jejich počet není omezen. Jakákoliv nenulová hodnota značí konec busy signálu a připravenost karty na další příkaz.

6.2.3 Odpověď R2

Odpověď R2 má velikost 2 byty a je odeslána jako odpověď na příkaz SEND_STATUS. Jednotlivé bity jsou popsány v tabulce 6.3.

Tab. 6.3 - Formát odpovědi R2

1. byte								2. byte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	Parameter error	Address error	Erase sequence error	Com CRC error	Illegal command	Erase reset	In idle state	Out of range csd overwrite	Erase param	Wp violation	Card ECC failed	CC error	Error	Wp erase skip lock/unlock cmd failed	Card is locked

Obsah prvního bytu je totožný jako u odpovědi R1, hodnoty logické jedničky v bitech druhého bytu značí:

- Erase param: neplatný výběr pro mazání,
- Write protect violation: příkaz se pokouší zapsat do bloku chráněného proti přepsání,
- Card ECC failed: vnitřní kontrola a oprava chyb byla spuštěna, ale selhala,
- CC error: chyba vnitřního řadiče karty,
- Error: nastala obecná nebo neznámá chyba,
- Write protect skip | lock/unlock command failed: příkaz se pokouší smazat chráněný sektor nebo nastala chyba při zamykání/odemykání karty,
- Card is locked: karta zamčena proti zápisu.

6.2.4 Odpověď R3

Má délku 5 bytů (tab. 6.4), první byte je stejný jako R1, další čtyři obsahují hodnotu OCR (Operation Conditions Register - registr provozních podmínek) registru.

Tab. 6.4 - Formát odpovědi R3

39								32	31								0
'0'	x	x	x	x	x	x	x	'xx...xx'									
R1								OCR									

6.2.5 Odpovědi R4 a R5

Tyto druhy odpovědi jsou rezervovány pro použití u SDIO zařízení.

6.2.6 Odpověď R7

Přichází po příkazu SEND_IF_COND (provozní stav karty). Má délku 5 bytů (tab. 6.5), první byte je opět stejný jako R1.

Tab. 6.5 - Formát odpovědi R7

39		32	31		28	27		12	11		8	7		0
R1			Command version			Reserved bits			Voltage accepted			Check pattern		

6.3 Inicializace SD karty do SPI režimu

Na počátku komunikace je SD paměťová karta probuzena v režimu SD. Proto je nutné nejprve kartu přepnout do režimu *SPI*. Inicializační sekvence se skládá z několika kroků. Po přivedení napětí na vstup je potřeba vyčkat alespoň 1 milisekundu. Hodiny musí být v tuto chvíli nastaveny na hodnotu v rozmezí 100 kHz až 400 kHz.

Přepnutí začne odesláním příkazu pro softwarový reset *CMD0* a současným uvedením výstupu *SS* do logické nuly. Hodnota prvního bytu v šestnáctkové soustavě je *0x40h*. Protože resetovací příkaz nemá žádné argumenty, ale délka je dána, jsou odeslány 4 byty logických nul (*0x00h*). Následuje hodnota kontrolního součtu s hodnotou *0x95h*. Kontrolní součet je sice u SPI protokolu vypnut, ale před přepnutím do SPI režimu je potřeba ho použít. V tuto chvíli můžeme na výstup *MOSI* odesílat byty plné jedniček (nic neznamenající byty) a kontrolovat vstup *MISO*. Ve chvíli, kdy se na něm objeví hodnota *0x01h*, je karta inicializována do SPI režimu. V tuto chvíli můžeme hodiny nastavit na rychlost, kterou SPI protokol u SD karty dovolí, tedy maximálně na 25 MHz.

6.4 Režim přenosu

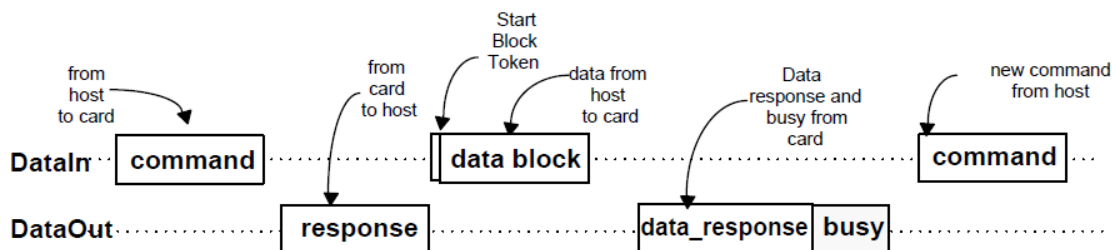
Po dokončení inicializace přejde SD karta do Stand-By režimu. Odesláním příkazu *CMD7* dojde k vybrání určité karty a jejímu přepnutí do přenosového režimu. V našem případě, kdy je připojena pouze jedna karta, dojde k přepnutí do Transfer režimu automaticky. V něm je již možné pomocí příkazů *CMD24* a *CMD17* zapisovat a číst data.

6.5 Nastavení velikosti bloku dat

Před započítím komunikace je potřeba nastavit velikost bloku dat, který bude zapisován nebo čten. U karet SDHC a SDXC je tato hodnota nastavena na velikost 512 bytů, u karet SDSC je možné ji nastavit. Příkaz pro nastavení je *CMD16*. Argumentem je hodnota velikosti bloku.

6.6 Zápis na SD kartu

Po přechodu karty do přenosového režimu odešleme příkaz CMD24, kde argumentem je adresa bloku, na kterou chceme data zapsat. Karta odešle odpověď R1. Poté odešleme celý blok dat, jehož velikost mohla být nastavena pomocí příkazu CMD16. Každý blok dat začíná startovacím tokenem o délce jednoho bytu. Po přenesení celého bloku je kartou odeslána odpověď s výsledkem přenosu. Říká, zda byl v pořádku kontrolní součet (pokud je používán) nebo zda nenastaly některé obecné chyby zápisu. Po této odpovědi začne programovat data do karty a na datovou linku odesílá tokeny značící zaneprázdnění. Po dokončení naprogramování dat na kartu je možné pomocí příkazu CMD13 zkontrolovat status přenosu. Některé chyby jsou totiž detekovatelné až při programování dat na kartu (např. adresa mimo rozsah, ochrana proti zápisu). Posloupnost přenosu zobrazuje obrázek 6.1.



Obr. 6.1 - Posloupnost zápisu na SD kartu, převzato z [1]

6.6.1 Data response token

Po každém bloku dat odeslaných na kartu je vytvořen token dlouhý jeden byte (tab. 6.6). Status bity mohou mít hodnoty:

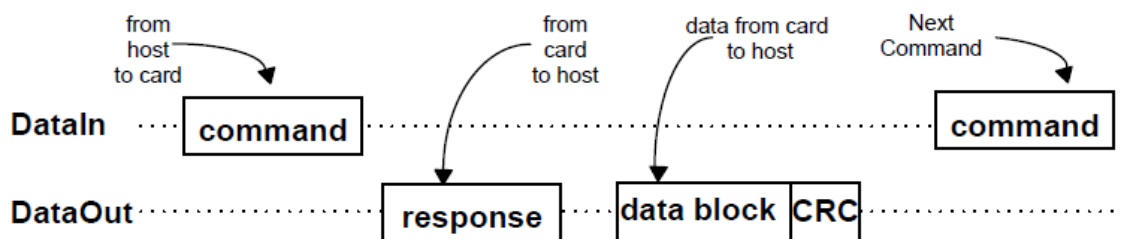
- '010' - data přijata,
- '101' - data nepřijata, chyba CRC,
- '110' - data nepřijata, chyba zápisu.

Tab. 6.6 - Formát Data response tokenu

7	6	5	4	3	2	1	0
x	x	x	'0'	Status		'1'	

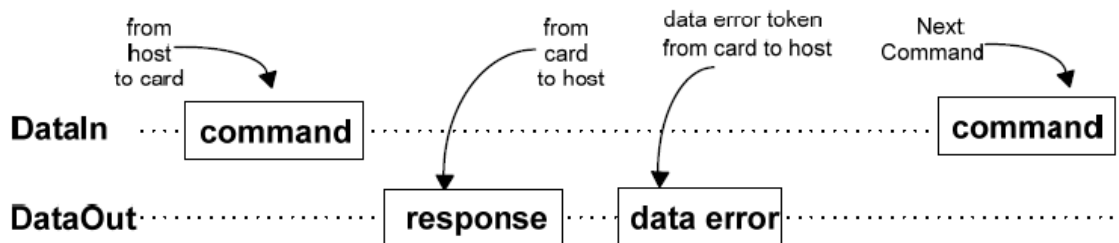
6.7 Čtení z SD karty

Pokud je karta v přenosovém stavu, můžeme příkazem CMD17 přečíst data o velikosti bloku nastaveného pomocí CMD16. Argumentem je adresa dat. Po obdržení příkazu karta odešle odpověď typu R1. Následuje blok dat z dané adresy doplněný o kontrolní součet CRC. V případě, že není používán, je opět nahrazen stejně dlouhými daty. Po odeslání celého bloku očekává karta další příkaz. Posloupnost přenosu ukazuje obrázek 6.2.



Obr. 6.2 - Posloupnost čtení z SD karty, převzato z [1]

V případě, že dojde k chybě, karta neodešle žádná data. Je odeslán speciální chybový token (obr. 6.6).



Obr. 6.3 - Chyba při čtení z SD karty, převzato z [1]

6.7.1 Chybový token při chybě čtení

Token má velikost 1 byte (tab. 6.7) a udává, k jaké chybě došlo.

Tab. 6.7 - Formát chybového tokenu při chybě čtení

7	6	5	4	3	2	1	0
'0'	'0'	'0'	'0'	x	x	x	x

Hodnotou logické jedničky značí bit číslo 3, že nastala chyba. Bit číslo 2 ukazuje na chybu vnitřního řadiče karty, číslo 1 na chybu interní kontroly a opravy chyb (ECC). Bit 0 značí adresu mimo rozsah karty.

6.8 Startovací token při přenosu dat

Po příkazech pro čtení či zápis dat jsou data odeslána po datových tokenech. Data jsou přenášena od nejvýznamnějšího bitu (MSB). Mohou být dlouhá od 4 do 515 bytů.

- 1. byte: start blok, má pevně danou hodnotu (tab. 6.8),

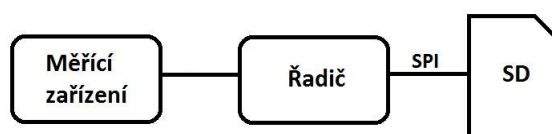
Tab. 6.8 - Formát start bytu ve startovacím tokenu

7	6	5	4	3	2	1	0
'1'	'1'	'1'	'1'	'1'	'1'	'1'	'0'

- Byty 2 až 513 (v závislosti na velikosti bloku): uživatelská data,
- Poslední dva byty: kontrolní součet CRC.

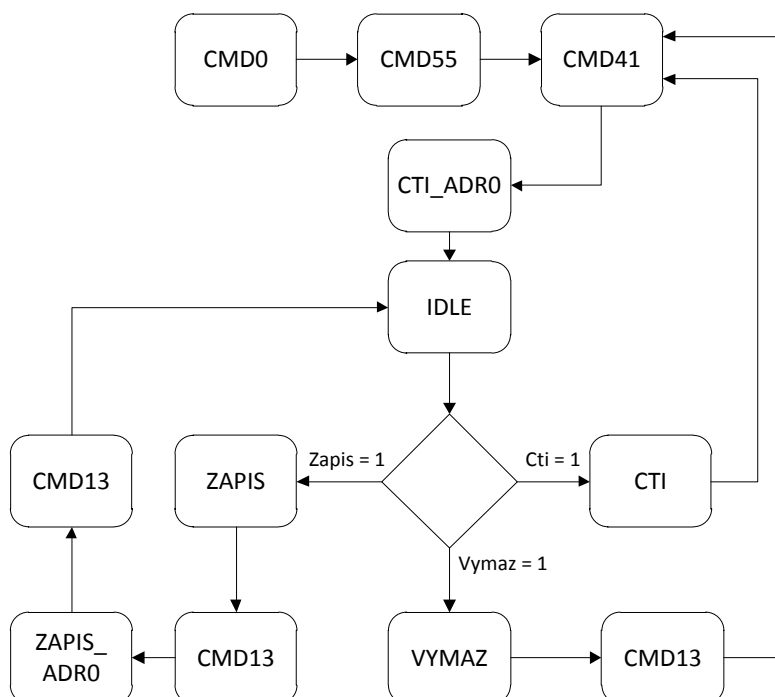
7 Řadič paměťové karty

Řadič je zapojen mezi měřicím obvodem a paměťovou kartou. S SD kartou komunikuje pomocí SPI sběrnice, s měřicím zařízením může komunikovat po různých sběrnicích. Pro tuto práci bylo měřicí zařízení nahrazeno obvodem, na kterém pomocí kódového spínače nastavíme počáteční hodnotu a ta je vždy zvětšena o předem danou hodnotu a nahrána. Takto se vygeneruje sada dat o velikosti bloku dat SD karty. Zjednodušené zapojení je vidět na obrázku 7.1.



Obr. 7.1 - Schéma zapojení řadiče

Celý řadič se dá ještě rozdělit na stavové automaty pro inicializaci karty, pro zjištění nejbližší volné adresy, pro zápis, čtení a vymazání. Obecný automat je na obr. 7.2. Nejdříve dojde k resetu a inicializaci karty, dále k načtení hodnoty volné adresy. Ve stavu IDLE automat čeká na pokyny zápisu, čtení či vymazání.



Obr. 7.2 - Obecný stavový automat komunikace s SD kartou

Původní záměr byl napsat řadič tak, že každý příkaz bude mít svůj stavový automat. Postupně ale došlo k zjednodušení kódu, kdy nebylo potřeba některé zcela identické stavy neustále v kódu opakovat. Proto byl přidán signál *navr_stav*, pomocí kterého je automat schopen se z těchto „univerzálních“ stavů přesunout do správného stavu.

Kromě stavových automatů je v řadiči ještě přítomna přepínatelná dělička frekvence a obvod pro zpoždění začátku inicializace.

7.1 Stavový automat inicializace SD karty

Na obrázku 7.3 je stavový automat, kterým je inicializována SD karta. Do stavu RESET se automat dostane až po přechodu signálu *radic_en*, generovaného obvodem opožděného startu po uplynutí minimálně 1 ms, do logické 1. Výstup *SS* sběrnice SPI je nastaven do logické 1. Ve stavu INIT je odesláno 80 hodinových pulzů na vodič *SCLK* a po jejich odeslání je *SS* nastaven do logické 0. Automat přejde do stavu CMD0, ve kterém je do vnitřního signálu řadiče *sig_cmd* nastavena hodnota příkazu "FF400000000095" v šestnáctkové soustavě. Po prvním bytu s hodnotou "FF" následuje byte obsahující startovací a přenosový bit a hned po nich 6 bitů s hodnotou příkazu. V tomto stavu jde o CMD0, tedy reset SD karty, proto je tato hodnota 0. Počet odesílaných bitů příkazu je 56, proto je tato hodnota nastavena do čítače bitů. Návratový stav je nastaven na CMD55 a celý automat přejde do stavu POSLI_CMD.

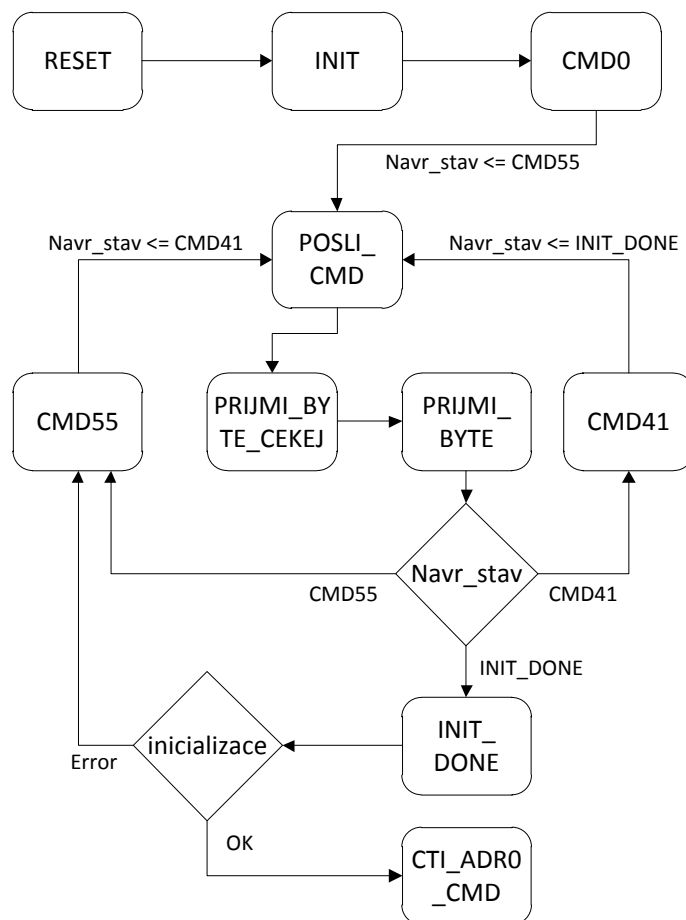
Tento stav je společně s PRIJMI_BYTE_CEKED a PRIJMI_BYTE stejný i pro automaty zjištění volné adresy, zápisů a čtení (viz kapitoly 7.2 až 7.6). Jelikož jsou v každém z nich vcelku hojně využívány, jsou vždy na obrázku uvedeny znovu, přičemž funkce je naprosto stejná a v kódu jsou popsány pouze jednou. Stav, ze kterého do POSLI_CMD přejdeme, vždy musí nastavit hodnotu signálu *navr_stav*, aby bylo možné následně ze stavu PRIJMI_BYTE pokračovat v činnosti automatu.

Ve stavu POSLI_CMD je postupně na výstup *MOSI* odesláno všech 56 bitů příkazu a poté automat přejde do stavu PRIJMI_BYTE_CEKED. Zde je nastaven počet bitů očekávané odpovědi karty. V tomto případě, kdy se jedná o odpověď na příkaz, je počet přijímaných bitů nastaven na 8. Ve stavu PRIJMI_BYTE je pak těchto 8 bitů postupně přijato a zapsáno na výstup *data_out*. Po PRIJMI_BYTE následuje stav podle aktuální hodnoty signálu *navr_stav*, tedy CMD55.

Stav CMD55 je takřka totožný jako CMD0, liší se pouze hodnotou příkazu. Příkaz 55 sděluje kartě, že po něm bude následovat aplikačně specifický příkaz. Následuje stav POSLI_CMD a stejný postup jako u CMD0. Návratovým stavem z PRIJMI_BYTE je tentokrát CMD41.

Stejně jako v předchozích dvou případech, kdy se odesílaly příkazy, je nastaven jejich argument (např. pro CMD0 jde o hodnotu 0, pro CMD55 o hodnotu 55) a pomocí POSLI_CMD je celý příkaz odeslán. Návratový stav je INIT_DONE.

Ve stavu INIT_DONE se zkontroluje poslední přijatý byte a pokud je hodnota nultého bitu rovna logické 0, inicializace proběhla v pořádku a automat pokračuje do stavu CTI_ADR0_CMD. Pokud ne, je opět odeslán CMD55 a po něm CMD41, dokud neproběhne inicializace v pořádku.



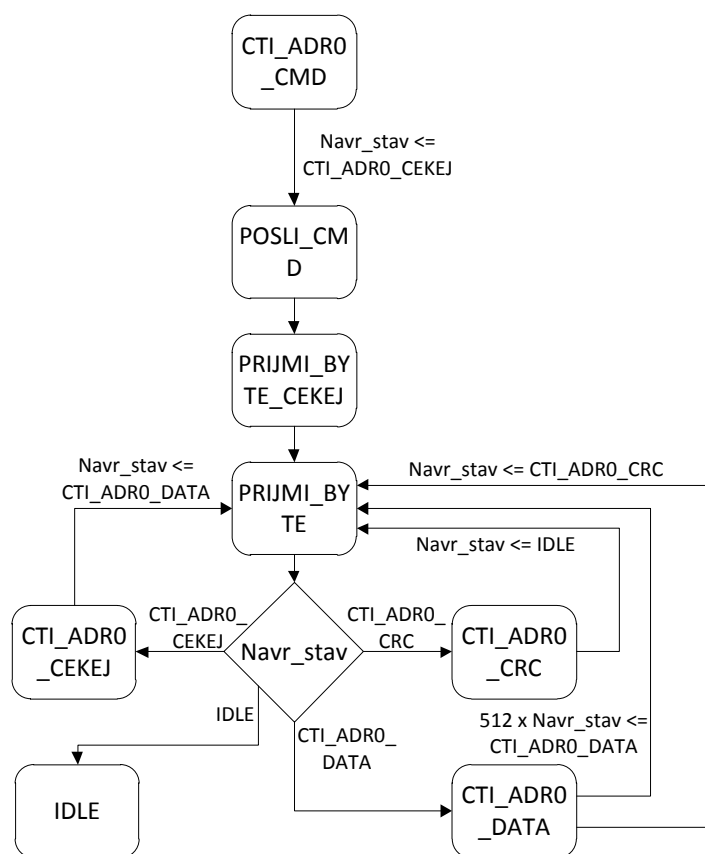
Obr. 7.3 - Stavový automat pro inicializaci SD karty

7.2 Stavový automat pro zjištění nejbližší volné adresy

Schéma stavového automatu pro zjištění nejbližší volné adresy je na obrázku 7.4. Jedná se v podstatě o jednoduchý souborový systém, kdy je na nejnižší možné (pevně dané) adrese karty uložena hodnota ADR0, která odkazuje na adresu, kam je možné zapisovat. Tato hodnota je načtena hned po inicializaci karty, aby byla k dispozici pro zápis, případně pro čtení.

Automat inicializace karty skončil ve stavu CTI_ADR0_CMD. Tam začíná tento automat. Opět nastavíme do signálu řadiče *sig_cmd* hodnotu příkazu, tentokrát pro přečtení jednoho bloku dat (CMD17). Je nastavena nulová adresa a počet bitů 55. Jde o startovací byte plný logických 1 a 6 bytů příkazu (viz kapitola 6.1).

Pokračuje se opět do POSLI_CMD, PRIJMI_BYTE_CKEJ a PRIJMI_BYTE. Ve stavu CTI_ADR0_CKEJ řadič počká na přechod karty ze zaneprázdněného stavu, který je signalizován hodnotou logické 1 na vodiči *MISO*. Následuje stav PRIJMI_BYTE, kde je postupně po jednotlivých bytech přijat celý blok dat. Při čtení hodnoty adresy nás zajímá pouze 5., 4., 3. a 2. byte, protože dva poslední byty obsahují kontrolní součet (u SPI režimu libovolná hodnota). Po přijetí posledních dvou bytů následuje stav IDLE a hodnota nejbližší volné adresy je uložena v registru, jehož výstupem je signál *volna_adresa*.



Obr. 7.4 - Stavový automat pro zjištění nejbližší volné adresy

7.3 Stavový automat zápisu na SD kartu

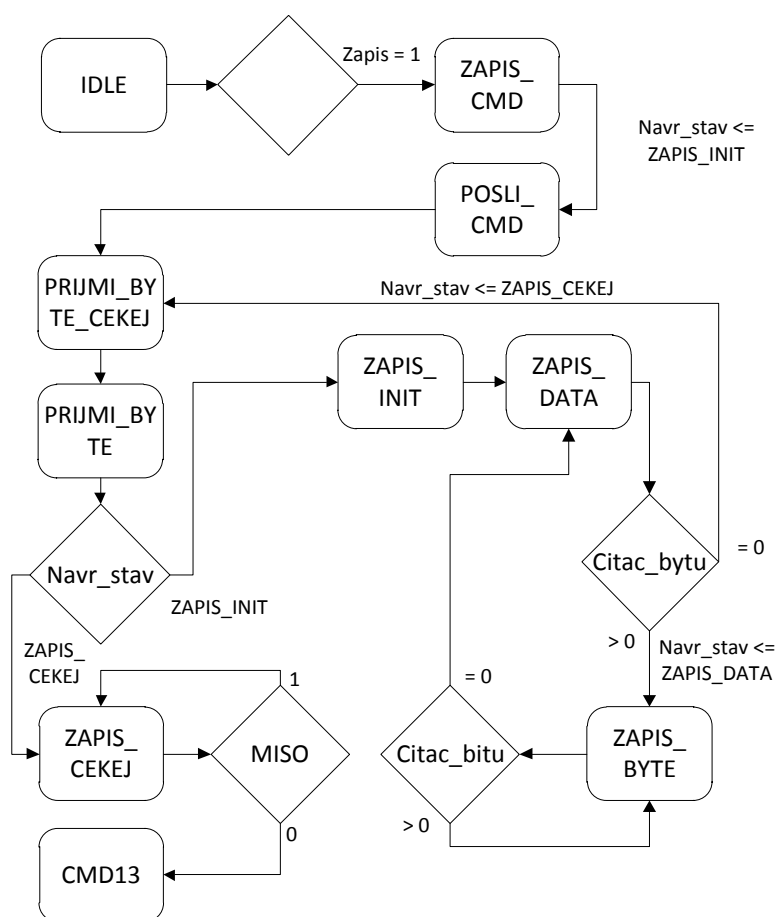
Stavový automat popisující zápis na SD kartu je na obrázku 7.5. Dosažení stavu IDLE dá automat najevo změnou výstupu *idle_done* do logické 1. To je důležité pro děličku frekvence, která pozná, že může frekvenci hodinového signálu zvýšit až na 25 MHz, což je mezní frekvence SPI sběrnice.

V tomto stavu může automat dostat tři různé příkazy - zapiš, čti nebo vymaž. V případě, že je vstup *zapis* v logické 1, přejde automat do stavu ZAPIS_CMD.

Signál *sig_cmd* se zde nastaví na hodnotu příkazu 24, což je zapsání jednoho bloku dat. Ve stavu ZAPIS_CMD se návratový stav změní na ZAPIS_INIT a přejde do stavu POSLI_CMD.

Pomocí následujících dvou stavů PRIJMI_BYTE_CEKJ a PRIJMI_BYTE přijme řadič odpověď karty a skrz stav ZAPIS_INIT začne odesílat data k zapsání na kartu.

Ve stavu ZAPIS_DATA se postupně odešle startovací byte s hodnotou "FE", poté je nastaven výstup řadiče *dalsi_byte* do logické 1, čímž je dán signál, že řadič očekává data. Po příchodu signálu na vstup *platna_data_in* je odeslán byte dat ze vstupu *data_in*. To se opakuje pro všech 512 bytů dat. Nakonec jsou odeslány 2 byty CRC (viz kapitola 6.8). Poté je přijata odpověď karty a následuje stav ZAPIS_CEKJ, kde automat čeká na logickou 0 na vstupu *MISO*. Když přijde, následuje kontrola zapsaných dat.



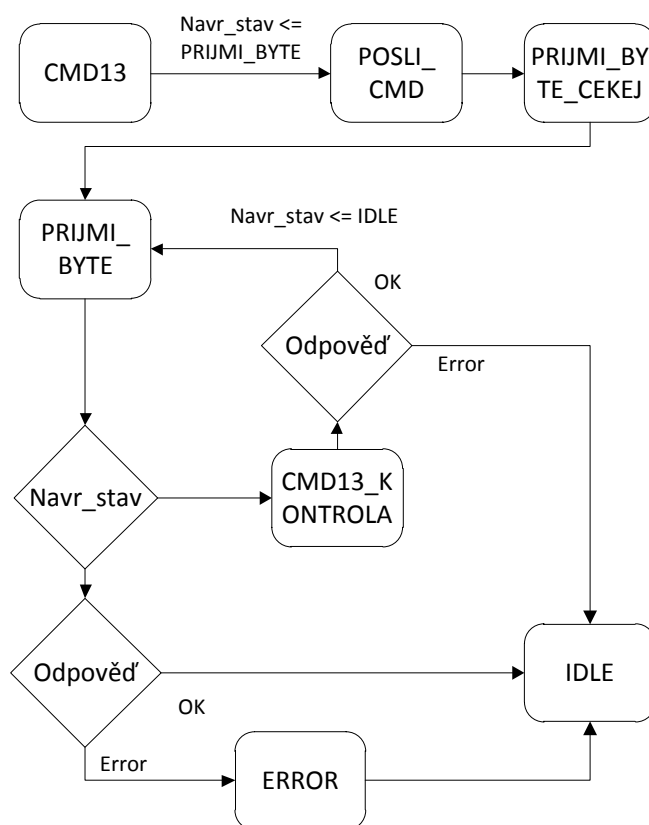
Obr. 7.5 - Stavový automat zápisu na SD kartu

7.4 Stavový automat kontroly zapsání dat

Kontrola zapsání dat se provádí pomocí příkazu CMD13, stavový automat ukazuje obrázek 7.6. Jde o vyžádání si obsahu Status registru karty. Postup je stejný jako v předchozích případech, kdy se odesílal příkaz. Rozdíl je v tom, že na příkaz CMD13 je očekávána odpověď typu R2 (viz kapitola 6.2.3), tedy o 1 byte delší, než R1. Oba byty jsou zkontrolovány. Pokud nastala chyba, přejde automat do stavu ERROR, kde je nastavena signalizace chyby na výstup pomocí LED.

Ze stavu ERROR přejde automat zpět do stavu IDLE. Zde je možné data znovu uložit, což musí řešit zařízení na vstupu.

Tato kontrola proběhne po jakémkoliv zapsání na kartu, tedy i u aktualizace volné adresy či vymazání.



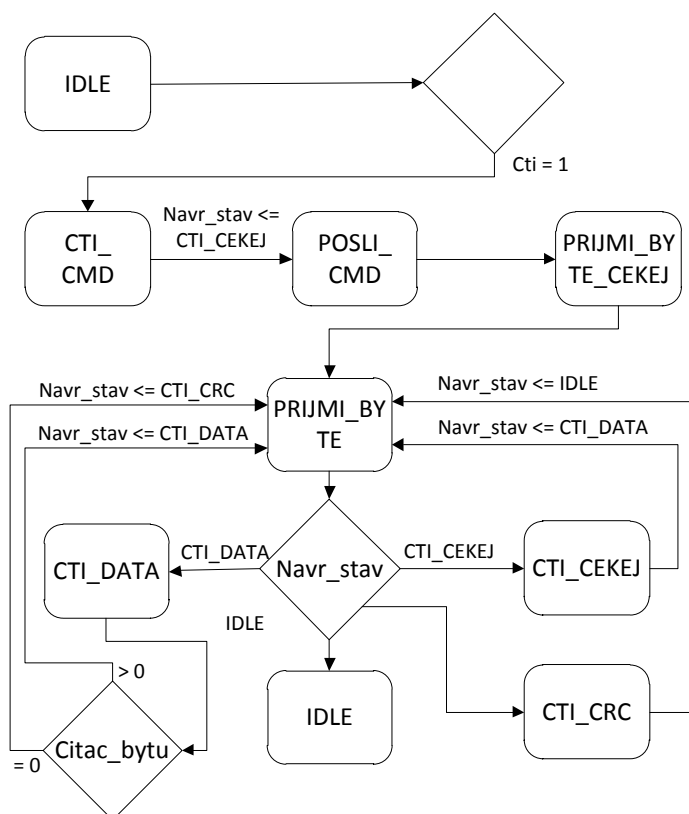
Obr. 7.6 - Stavový automat kontroly zapsání dat

7.5 Stavový automat aktualizace volné adresy

Stavový automat pro aktualizaci je v podstatě stejný jako pro zápis, rozdíl je ve stavu ZAPIS_ADR0_CMD (ZAPIS_CMD u zápisu), kde je adresa nastavena napevno na nulovou hodnotu, a v ZAPIS_ADR_DATA, kde nejsou posílána data ze vstupu, ale do posledních čtyřech bytů je uložena hodnota volné adresy zvětšená o počet bytů odesílaných do karty. V tomto případě jde o 512 bytů. Hodnota nové adresy je změněna i v registr, který je představován signálem *volna_adresa*.

7.6 Stavový automat čtení z SD karty

Automat (obr. 7.7) opět začíná ve stavu IDLE, kde po změně vstupu *cti* do logické 1 přejde do stavu CTI_CMD. Tam je nastavena hodnota CMD17, přečti jeden blok dat. Pokud není hodnota signálu *volna_adresa* nulová, do argumentu se zapíše zmenšená o 512. Tím se přečtou data z naposledy zapsané adresy.



Obr. 7.7 - Stavový automat čtení z SD karty

Automat čtení pracuje stejně jako automat pro čtení volné adresy. Po odeslání příkazu přijímá data ze vstupu *MISO*. Na rozdíl od předchozího automatu ale přijatá data nejsou zapisována do registru adresy, ale postupně do registru signálu *data_komplet*. Po obdržení celého bloku je nastaven výstup *data_ready* do logické 1. Tím se dá najevo, že si je výstupní obvod může načíst. Jsou tam uložena kompletní přijatá data, tedy i s úvodním bytem plným logických 1 i s dvěma byty CRC. Tento způsob uložení je zvolen hlavně z důvodu lepší kontroly přijatých dat.

7.7 Dělička frekvence

Dělička frekvence dělí vstupní frekvenci oscilátoru desky, která má hodnotu 32 MHz, na potřebné kmitočty. Ty jsou dva a vybírá si je v podstatě sám řadič. Protože řadič ještě frekvenci výstupních hodin pro sběrnici SPI dělí dvěma, jsou hodnoty jeho řídicích hodin dvojnásobné.

Začíná se vždy frekvencí 400 kHz, která je nastavena po celou dobu inicializace. Ve chvíli, kdy chce začít řadič načítat volnou adresu, je dělička přepnuta na výstupní frekvenci 32 MHz, tedy vstup jde přímo na výstup. Tato frekvence již zůstane nastavena až do případného resetu řadiče nebo vypnutí a zapnutí.

7.8 Zpoždění začátku inicializace

Obvod zajišťuje potřebné minimálně jednomilisekundové zpoždění začátku inicializace. Jedná se o jednoduchý čítač, který počítá počet hodinových impulzů na vstupu *clk_sd* a jehož maximem je 110. Po dosažení maximální hodnoty je výstup obvodu *delay_out* nastaven do logické 1 a je spuštěn automat řadiče. Obvod je navržen tak, aby bylo minimální zpoždění zaručeno v rozsahu vstupních frekvencí $100 \div 400$ kHz. Při vstupní frekvenci 400 kHz dojde ke zpoždění 4,4 ms, při frekvenci 100 kHz pak 1,1 ms.

7.9 Obvody vstupu a výstupu

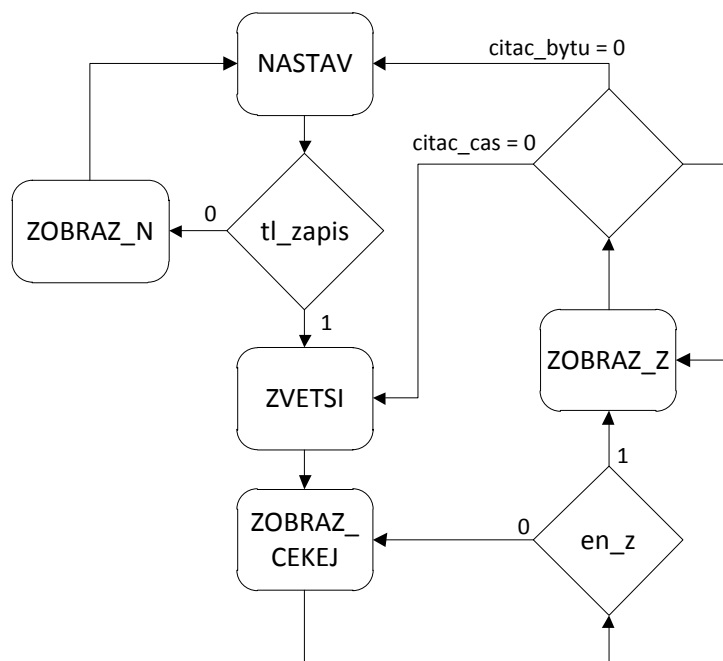
Součástí této práce jsou také obvody, které zajišťují vstup ukládaných dat a jejich zobrazení. Tato zařízení jsou navržena s ohledem na jednoduché odzkoušení celého řadiče nahraného na desce FUB. Řadič je na nich nezávislý, lze je tedy nahradit libovolnými obvody pro generování, respektive vyčítání dat.

7.9.1 Obvod vstupních dat

Tento obvod pracuje tak, že po uživatelském zadání úvodní hodnoty sám vygeneruje dalších 511 bytů dat. Každý následující byte je zvětšen o hodnotu 3. Tím docílíme dat, která všechna nemají stejnou hodnotu a při vyčtení uložených dat z karty můžeme jednoduše kontrolovat jejich správnost.

Hodnota se do obvodu zadává kódovým spínačem přítomným na desce. Zadávaná hodnota se průběžně zobrazuje na sedmisegmentových displejích DISPL1 a DISPL2. Po stisku tlačítka TL1 se zadaná hodnota uloží a do řadiče odešle na vstup *zapis* logikou 1. V okamžiku, kdy se na vstupu *en_z* objeví logická 1, začne generovat a odesílat data. Každý byte je následován logikou 1 na výstupu *platna_data_z*. Po odeslání všech 512 bytů je možné opět pomocí kódového spínače zadávat hodnotu.

Stavový automat řídicí obvod je na obr. 7.8. Dokud nepříjde příkaz zapsání, cyklicky čte hodnotu vstupu a zobrazuje ji na displejích. Ve stavu ZVETSI poté vždy hodnotu zvětší, opět jí zobrazí na displejích a odešle na výstup *data_out_z*.



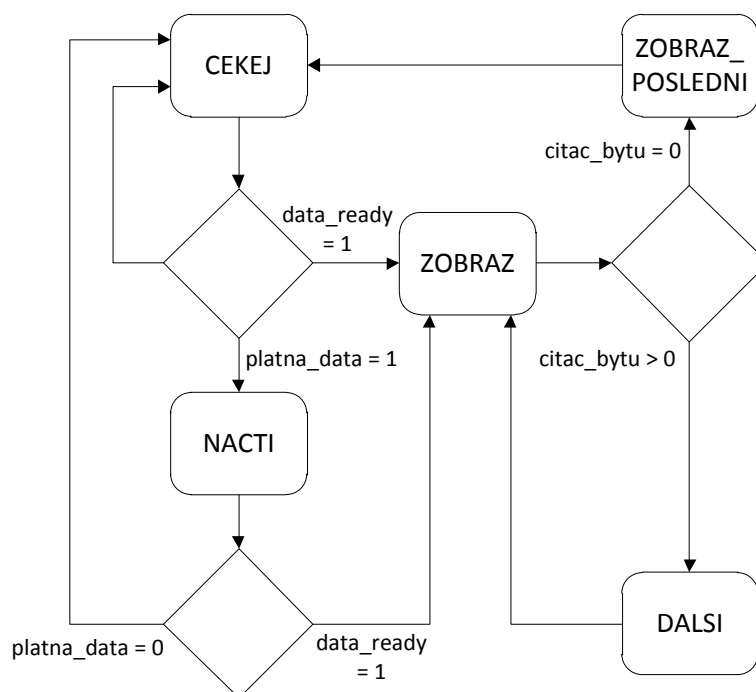
Obr. 7.8 - Stavový automat obvodu vstupních dat

7.9.2 Obvod výstupních dat

Jde v podstatě o jednoduchý řadič sedmissegmentového displeje. Na DISPL3 a DISPL4 zobrazuje hodnoty výstupu řadiče *data_out*.

Pomocí tlačítka TL2 je spuštěno čtení v řadiči a přeposílání přijatých dat na jeho výstup *data_out*. Vždy, když je zobrazen další byte, je výstup řadiče *platna_data_out* nastaven do logické 1. V tu chvíli se uloží hodnota vstupu obvodu výstupních dat *data_in_v* do signálu *data_komplet*. V okamžiku, kdy jsou data kompletní (výstup řadiče *data_ready_out* v logické 1), jsou hodnoty jednotlivých bytů postupně zobrazovány na displeji po dobu 0,5 sekundy.

Obvod je řízen pomocí stavového automatu na obr. 7.9. Při každé logické 1 na vstupu *platna_data* jsou vstupní data uložena. Po signalizaci kompletního odeslání dat řadičem přejde automat do režimu zobrazování a po posledním zobrazeném bytu zpět do stavu CEKEJ.



Obr. 7.9 - Stavový automat obvodu výstupních dat

8 Simulace a odzkoušení

Po navržení obvodu bylo nutné jej nejdříve odsimulovat v programu Quartus. Po vyladění nepřesností byl návrh nahrán na desku FUB a odzkoušen s reálnou paměťovou kartou. Celkové zapojení je na obrázku v příloze A.1.

8.1 Simulace

Nejdříve byly jednotlivě odsimulovány bloky řadiče, opoždění startu, změny frekvence a vstupní a výstupní obvody, následně i vše zapojené dohromady. Simulace probíhá bez reálné karty, její signály je tedy nutné nastavit ručně.

Nejprve proběhla simulace inicializace karty, kdy byly sledovány hodnoty *MOSI*, *SCLK* a *SS* sběrnice SPI. Tyto hodnoty byly zaznamenány a porovnány s požadovanými. Celá sekvence inicializace, tedy příkazy *CMD0*, *CMD55* a *CMD41*, následována příkazem pro čtení jednoho bloku *CMD17* pro přečtení volné adresy proběhla podle předpokladů. Přečtená data přesně odpovídala očekávaným hodnotám.

Simulace odesílání byla provedena tak, že po signálu logické 1 na vstup kompletního obvodu *tl_zapis* byla znovu sledována sběrnice SPI. Na výstupu *MOSI* byly sledovány hodnoty, které obvod odesílal do karty vždy, když přešel výstup *SCLK* do logické 1. Opět všechny hodnoty odesílané na sběrnici SPI odpovídaly zadaným. Sledován byl i výstup *chyba*, který podává informaci o chybě při zápisu. Tato chyba byla nastavena pomocí vstupu *MISO* ve chvíli, kdy byla přijímána odpověď na příkaz *CMD13*.

Simulování příjmu dat z karty byla složitější, bylo potřeba v přesně dané okamžiky nastavit vstup *MISO* do požadovaných hodnot. Tentokrát bylo nutné kromě sběrnice SPI sledovat také výstupy řadiče *data_out*, *platna_data_out* a *data_ready_out*. Nejdříve proběhla kontrola správnosti odesílaného příkazu a poté byl nastavován vstup *MISO* do hodnot, které by odesílala karta. Data odesílaná na výstup *data_out* byla stejná, jako hodnoty nastavené na vstupu *MISO*, řadič tedy přijímal data správně. Navíc při této simulaci byl odzkoušen i obvod výstupních dat, který po přijetí signálu o kompletnosti dat začal postupně nastřádané hodnoty odesílat na výstupy kompletního obvodu *seg3* a *seg4*.

8.2 Odzkoušení

Reálné odzkoušení proběhlo po přiřazení pinů a nahrání návrhu do desky FUB. K té byl přes dutinkovou lištu EXT 1 připojen slot paměťové karty.

Po signalizaci dokončení inicializace a načtení volné adresy pomocí rozsvícení LED 1 byla díky kódovému spínači DIP08 nastavena první hodnota, kterou jsme chtěli uložit na kartu. Po stisku tlačítka Tl 1 (zapiš) se postupně na sedmisegmentových displejích DISPL1 a DISPL2 zobrazilo 512 hodnot, které byly odeslány do řadiče k uložení na kartu.

Po opětovném rozsvícení LED 1 (stavový automat opět ve stavu IDLE) bylo stisknuto tlačítko Tl 2 (čti) a spuštěn režim čtení poslední zapsané adresy. Po opětovném dosažení stavu IDLE automatu řadiče začal obvod výstupních dat zobrazovat na dobu 0,5 sekundy na DISPL3 a DISPL4 jednotlivé zapsané hodnoty. Tentokrát jich ale zobrazil 515, protože součástí těchto dat byly i startovací a CRC byty. Všechna zobrazená data odpovídala hodnotám, uloženým v předchozím zápisu.

Odzkoušeno bylo i chování po stisku tlačítka Tl 3 (vymaž). Po jeho stisku a návratu do stavu IDLE (LED 1), bylo stisknuto tlačítko pro čtení, ale protože adresa byla nulová, na displejích se objevily nuly a automat řadiče zůstal ve stavu IDLE.

Dále následovalo dvakrát uložení dat, pokaždé s jinou počáteční hodnotou. Po nich opět čtení, které ukázalo na displejích hodnoty druhého zápisu, tedy z naposledy zapsané adresy.

Nakonec byl vyzkoušen i asynchronní reset, který celé zařízení uvede na začátek inicializace karty.

Závěr

Úkolem této práce bylo porovnat dostupné paměťové karty na trhu, vybrat z nich nejvhodnější pro jednoduché ukládání a čtení měřených dat. Dále bylo potřeba navrhnout v popisném jazyce VHDL řadič této karty, který by umožnil ukládání a čtení dat, a odzkoušet ho v reálném obvodu FPGA.

Z celkem velkého množství druhů paměťových karet na trhu se jako nejlepší řešení ukázaly paměťové karty Secure Digital. Jejich cena klesla na velmi přijatelnou úroveň, kapacitně také dostačují. Jejich největší výhodou oproti konkurenci byla specifikace řadiče. Jsou v ní popsány základní úlohy řadiče. Složitější, které ve volné specifikaci nejsou, týkající se např. zabezpečených částí karty, nebyly v této práci řešeny.

Řadič samotný ke své správné funkci potřebuje ještě děličku frekvence a obvod zpoždění startu. Samotný řadič je řízen pomocí stavového automatu, který nejdříve provede inicializaci karty a poté čeká na příkaz k zapsání, čtení nebo vymazání.

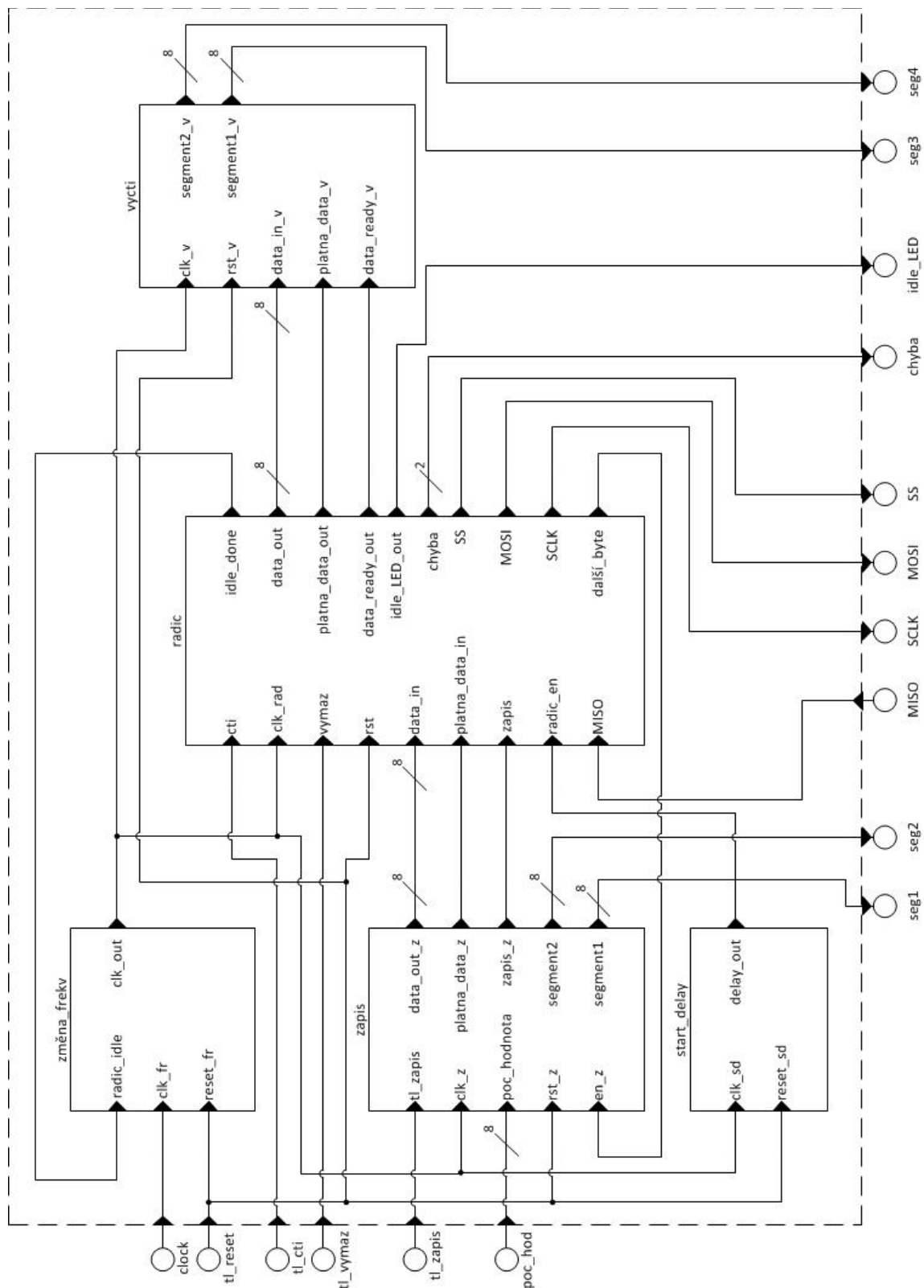
Data jsou do řadiče odesílána z obvodu vstupních dat, který je generuje od počáteční hodnoty nastavené ručně. Přečtená data jsou zobrazována na sedmisegmentovém displeji.

Bylo ověřeno správné zapsání a vyčtení dat, nejdříve pomocí počítačové simulace a následně i v reálném obvodu. Ověřena byla i další následující zapsání a vyčtení. Kdykoliv je možné celý řadič, včetně obvodů vstupu a výstupu přivést na začátek inicializace karty pomocí asynchronního resetu.

Řadič navržený v této práci je možné použít jako samostatný funkční blok v jakémkoliv obvodu FPGA. Zdrojové kódy v jazyce VHDL jsou na přiloženém CD.

Přílohy

Příloha A



A.1 - Celkové zapojení řadiče

Literatura

- [1] SD Specifications Part 1 Physical Layer Simplified Specification Version 3.01 [online], [cit. 23.3.2011],
URL:
<http://www.sdcard.org/developers/tech/sdcard/pls/simplified_specs/Part_1_Physical_Layer_Simplified_Specification_Ver3.01_Final_100518.pdf>
- [2] SD Specifications Part A2 SD Host Controller Simplified Specification Version 3.00 [online], [cit 23.3.2011]
URL:
<http://www.sdcard.org/developers/tech/pls/simplified_specs/Part_A2_SD_Host_Controller_Simplified_Specification_Ver3.00_Final_110225.pdf>
- [3] Žanta, M.: Návrh a realizace přípravku pro výuku hradlových polí.
[Bakalářská práce], FM TU, Liberec, 2009.
- [4] CF Cards [online], [cit 14.3.2011]
URL: <<http://compactflash.org/product-types/cf-cards/>>
- [5] Paměťové karty a jednotky USB od společnost Sony [online], [cit 14.3.2011]
URL: <<http://www.sony.cz/hub/pametove-karty-usb-jednotky>>
- [6] SD Standards Overview [online], [cit 15.3.2011]
URL: <<http://www.sdcard.org/developers/tech/>>